L7: Entry 3 of 9 File: USPT Jul 13, 1999

DOCUMENT-IDENTIFIER: US 5923859 A
TITLE: Dual arbiters for arbitrating access to a first and second bus in a computer system having bus masters on each bus

Abstract Text (1):
Arbitration circuitry in a computer system having a plurality of arbiters for arbitrating requests from bus masters on a PCI bus and an EISA bus. Each of the PCI and EISA buses have a plurality of masters. The PCI bus utilizes a modified LRU arbitration scheme, while the EISA bus utilizes a rotating priority scheme. The arbiter on the EISA bus includes a first level of arbitration and a second level of arbitration. The first level is assigned a plurality of requester types to determine the priority between the requestor types. Certain of the first level requestor types include a plurality of devices. If one of those certain requestor types wins priority on the first level arbitration cycle, a second level arbitration is performed to determine the priority between the plurality of devices.

Brief Summary Text (10):
The arbitration circuitry according to the present invention includes a first arbiter for arbitrating access to a first bus and a second arbiter for arbitrating access to a second bus. The first arbiter preferably utilizes a modified least recently used priority scheme for arbitrating requests from bus masters connected to the first bus as well as bus masters connected to the second bus. The second arbiter preferably Utilizes a rotating type priority scheme for arbitrating requests from bus masters connected to the second bus as well as from bus masters connected to the first bus. The second arbiter performs a first level rotating priority arbitration between a plurality of requestor types. Certain events disturb the first level rotating priority scheme. These events include the assertion of a non-maskable interrupt, which causes requests from bus masters connected to the second bus to be masked or bypassed. Certain of the requestor types include a plurality of devices or channels, requiring a second level of arbitration to be performed. For one requestor type, a second level rotating priority arbitration is performed. For a second requester type, a fixed or rotating priority arbitration is performed.

Detailed Description Text (12):
For the remainder of this description the CPU receives the [0] designation, the EISA bus E receives the [1] designation, the option connector 118 receives the [2] designation, the graphics connector 116 receives the [3] designation and the SCSI/NIC controller 120 receives the [4] and [6] designations, with [5] being unused. The output of the arbiter 402 is a series of signals referred to as the GNT [6:0] and SGNT[6:0] signals. The GNT signals are used to develop the CPUGNT.sub.--, EGNT.sub.-- and GNT.sub.-- [4:0] signals which are respectively the responses to the request signals provided to the request mask logic 400. The SGNT signals are the synchronized versions of the GNT signals, that is, they have been latched by a series of D-type flip-flops clocked on the PCICLK signal of the PCI bus P. The SGNT signals are provided to PCI status decode logic 404, which also receives the PCI control signals 104. Miscellaneous PCI cycle status signals are provided by the status decode logic 404. The SGNT signals are also provided to reservation and mask logic generally referred to as 406. As illustrated, the reservation and mask logic

406 includes two portions, a cycle timer 408, which receives two bits from an arbitrary I/O port, and reservation and mask generation logic 410, which receives one bit from an arbitrary I/O port. The output of the reservation and mask generation logic 410 is seven signals referred to as the MASK[6:0] signals, the priority masking signals, and seven signals referred to as the LOCKED[6:0] signals, which indicate which particular PCI bus master has locked the PCI bus P. Additionally, a signal referred to as RETRY.sub.-- MSTR or retry master is provided to indicate that a master has been aborted and a retry cycle has occurred.

Detailed Description Text (14):
In addition, the arbiter logic 402 receives the EREQ.sub.-- signal to determine if an EISA bus request is active. The EREQ.sub.-- signal is also provided to a new grant state machine 412, which is utilized to indicate when a new master can be granted control of the PCI bus P. Certain timers are associated with the grant phase, including the minimum grant timer 414 and a grant timeout timer 416. Sixteen bits of I/O from arbitrary ports are connected to the minimum grant timer 414, which also receives the GNT[0,2,3,6] signals and signals referred to as MIN.sub.-- GNT.sub.-- TMR.sub.-- STRT and MIN.sub.-- GNT.sub.-- TMR.sub.-- RST or minimum grant timer start and reset signals. The signals GNT[0,2,3,6] correspond to the CPU 100, the PCI options connector 118, the graphics connector 116, and the NIC controller, respectively. The minimum grant timer 414 produces two output signals referred to as the MIN.sub.-- TMR.sub.-- TO signal and the MIN.sub.-- GNT.sub.-- TO signals. Both of the signals indicate that the minimum grant timer 414 has timed out and that a new grant can occur.

Detailed Description Text (22):
As mentioned above, there are certain conditions when a PCI master is retried when referencing the memory. Then it is desirable to mask off all of the other request lines other than the one from the processor/main memory. It is desirable that this period be programmable because of varying processor speeds and other variables. To this end, two bits are utilized to define four options. The 00 value indicates that masking is disabled, while the other three combinations refer to 4, 6 and 8 PCICLK signal delays. These bits are provided to a CPU timer 470, with the CPU timer 470 being clocked by the PCICLK signal. The timer 470 is started upon receipt of a signal which is provided by the output of a five-input AND gate 472. The inputs to the AND gate 472 are the GNT[0] signal, the PCI.sub.-- IDLE signal, the CPU.sub.-- MSK[1] signal, the !CNTR.sub.-- ACTV signal and the !COUNT.sub.-- DISABLED signal. So when the timer 470 is not disabled, and not active, the PCI bus P is idle because of the retry, the EISA bus E is masked and the PCI bus P has been granted to the CPU, then the timer 470 is started by loading in the value indicated by the two bits from the arbitrary I/O port and counts down from the loaded value. The timer 470 is reset by the output of a two-input OR gate 474. One input to the OR gate 474 is the PCI.sub.-- RESET signal and the other input is provided by the non-inverting output of a D-type flip-flop 476. The D input of the flip-flop 476 receives the output of a two-input NOR gate 478 which receives at its inputs the two bits to define the time interval. The inverting output of the flip-flop 476 is the !COUNT.sub.-- DISABLED signal. The timer 470 is preferably a three bit timer and the three output bits are provided as three inputs to a three-input NOR gate 480 whose output is the !CNTR.sub.-- ACTV signal. The final output of the timer 470 is the CPU.sub.-- TO or CPU timeout signal which is provided when the timer 470 has counted down to 0 from the loaded value.

Detailed Description Text (63):
As noted, the relative priority of the requesters is determined by a rotating priority scheme between five requestor types. The priority of the various possible requesters is modified when a non-maskable interrupt or NMI is detected active and a PCI master is requesting the EISA bus E. In that case, requests from the DMA controller 180, the ISA bus masters, and the EISA bus masters are masked or bypassed in the current arbitration cycle. This allows the CPU 100 more time to process the NMI and reduces interrupt latency. When none of the request signals are

asserted, control of the EISA bus E defaults to one of the PCI masters to shorten the CPU access to the EISA bus E when needed. As noted above with respect to the PCI arbiter 184, the CPU 100 is parked on the PCI bus P when none of the other PCI bus masters are requesting the bus. Thus, on a cache miss, in non-cacheable cycles, or in I/O cycles, CPU access to the EISA bus E is sped up.

Detailed Description Text (102):
Thus, a computer system has been described having a plurality of arbiters for arbitrating requests from bus masters on a PCI bus and an EISA bus. Each of the PCI and EISA buses have a plurality of masters. The PCI bus utilizes a modified LRU arbitration scheme, while the EISA bus utilizes a rotating priority scheme. The arbiter on the EISA bus includes a first level of arbitration and a second level of arbitration. The first level is assigned a plurality of requestor types to determine the priority between the requester types. Certain of the first level requestor types include a plurality of devices. If one of those certain requestor types wins priority on the first level arbitration cycle, a second level arbitration is performed to determine the priority between the plurality of devices.

CLAIMS:

1. An arbitration circuit in a computer system having a first bus and a second bus, a plurality of first bus masters connected to the first bus, and a plurality of second bus masters connected to the second bus, the plurality of first bus masters providing a corresponding number of first bus request signals for the first bus, the plurality of second bus masters providing a corresponding number of second bus request signals for the second bus, the arbitration circuit comprising:

a first arbiter for connection to the first bus, said first arbiter responsive to the plurality of first bus request signals for providing a first-to-second signal indicating a request from a first bus master for the second bus, said first arbiter including:

a first prioritizer responsive to the plurality of first bus request signals and a second-to-first request signal for determining the highest priority bus master on the first bus; and

a first granting circuit coupled to said first prioritizer for granting ownership of the first bus to the highest priority bus master; and

a second arbiter for connection to the second bus, said second arbiter responsive to the plurality of second bus request signals for providing the second-to-first request signal indicating a request from a second bus master for the first bus, said second arbiter including:

a first level arbiter responsive to the first-to-second request signal and the plurality of second bus request signals for performing a first level arbitration to determine priority between a plurality of requester types, said plurality of requestor types including a first requestor type comprising the first bus masters represented by said first-to-second request signal, the other plurality of requestor types each containing a different portion of the plurality of second bus masters;

a second level arbiter coupled to said first level arbiter for performing a second level arbitration if one of certain of said requestor types is determined to have the highest priority, said second level arbiter for determining which of the plurality of second bus masters in said highest priority requester type has the highest priority; and

a second granting circuit coupled to said first level arbiter and said second level

arbiter for granting ownership of the second bus to the highest priority bus master.

17. A computer system, comprising:

a first bus;

a second bus;

a plurality of first bus masters connected to said first bus, wherein said plurality of first bus masters provide a corresponding number of first bus request signals for said first bus;

a plurality of second bus masters connected to said second bus, wherein said plurality of second bus masters provide a corresponding number of second bus request signals for said second bus;

a first arbiter coupled to said first bus, said first arbiter responsive to said plurality of first bus request signals for providing a first-to-second request signal indicating a request from a first bus master for said second bus, said first arbiter including:

a first prioritizer responsive to said plurality of first bus request signals and a second-to-first request signal for determining the highest priority bus master on said first bus; and

a first granting circuit coupled to said first prioritizer for granting ownership of said first bus to the highest priority bus master; and

a second arbiter coupled to said second bus, said second arbiter responsive to said plurality of second bus request signals for providing the second-to-first request signal indicating a request from a second bus master for said first bus, said second arbiter including:

a first level arbiter responsive to said first-to-second request signal and said plurality of second bus request signals for performing a first level arbitration to determine priority between a plurality of requestor types, said plurality of requester types including a first requester type comprising said first bus masters represented by said first-to-second request signal, said other plurality of requester types each containing a different portion of said plurality of second bus masters;

a second level arbiter coupled to said first level arbiter for performing a second level arbitration if one of certain of said requestor types is determined to have the highest priority, said second level arbiter for determining which of said plurality of second bus masters in said highest priority requestor type has the highest priority; and

a second granting circuit coupled to said first level arbiter and said second level arbiter for granting ownership of said second bus to the highest priority bus master.